CS3205: Introduction to Computer Networks

Major Assignment-2: Implementing a HTTP Proxy

By Akash NA(CS17B020) and Vamsi KV(CS17B045)

March 9, 2020

1) Introduction and Implementation Design:

a) Taking command line input of the proxy port number.

b) Setup the Proxy-Client socket in the parent proxy process.

c) The parent routine where the current parent process listens on the port and accepts connections from clients and forks into children to handle those clients.

d) The child routine which handles the client is as follows:

i) It reads the request from the client into a string format.

ii) It parses it for errors, it also checks if it is a valid GET request and not a POST/Other HTTP method request. It notifies user of the particular error accordingly. (Bad Request (400) and Not implemented (501)).

iii) Once the validation is completed, we parse the request and process it so that the required fields are modified and additional header information is appended/modified to the original request so that it is now suitable to be sent to the URL server.

iv) Setup a Proxy-Server socket in the child proxy process.

v) Forward the HTTP GET request to the URL Server.

vi) Read the reply from the URL Server and forward it to the Client.

e) Make sure unnecessary file descriptors are closed, especially before termination.

2) Experiments and Observations:

a) Single Client testing

```
File  Edit  View  Search  Terminal  Help
akhash@akhash:~/A3_1.1$ telnet www.cse.iitm.ac.in 80
Trying 10.6.8.2...
Connected to cse.iitm.ac.in.
Escape character is '^]'.
GET http://www.cse.iitm.ac.in/ HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 09 Mar 2020 06:19:16 GMT
Server: Apache/2.4.10 (Debian)
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or
g/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="EN" lang="EN" dir="ltr">
<head profile="http://gmpg.org/xfn/11">
<title>Department of Computer Science & Engineering</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta http-equiv="imagetoolbar" content="no" />
```

Figure 2.1: This figure is the Direct testing of the www.cse.iitm.ac.in website.

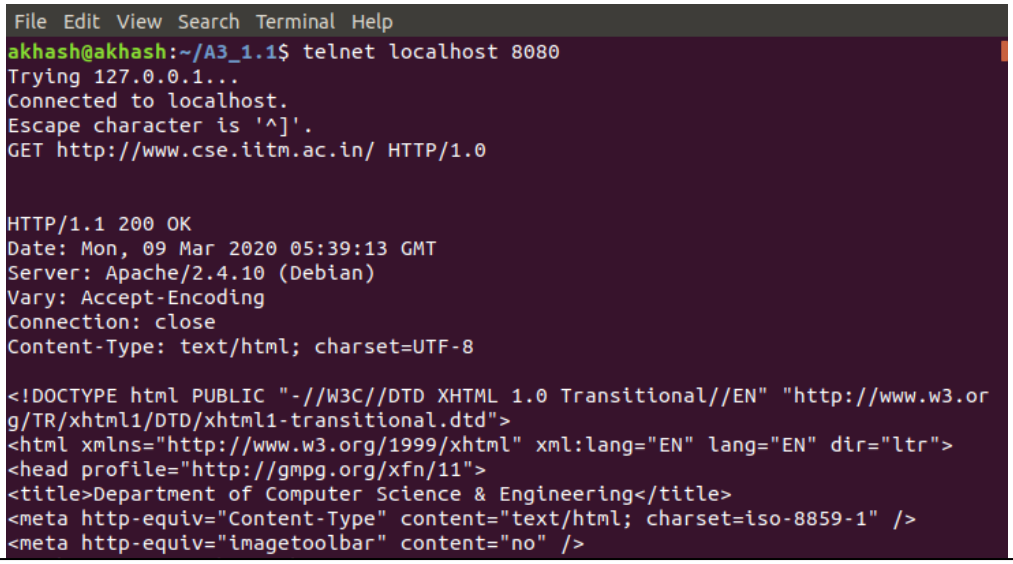


Figure 2.2: This figure is the Proxy testing of the www.cse.iitm.ac.in website.

b) Concurrent proxy testing:

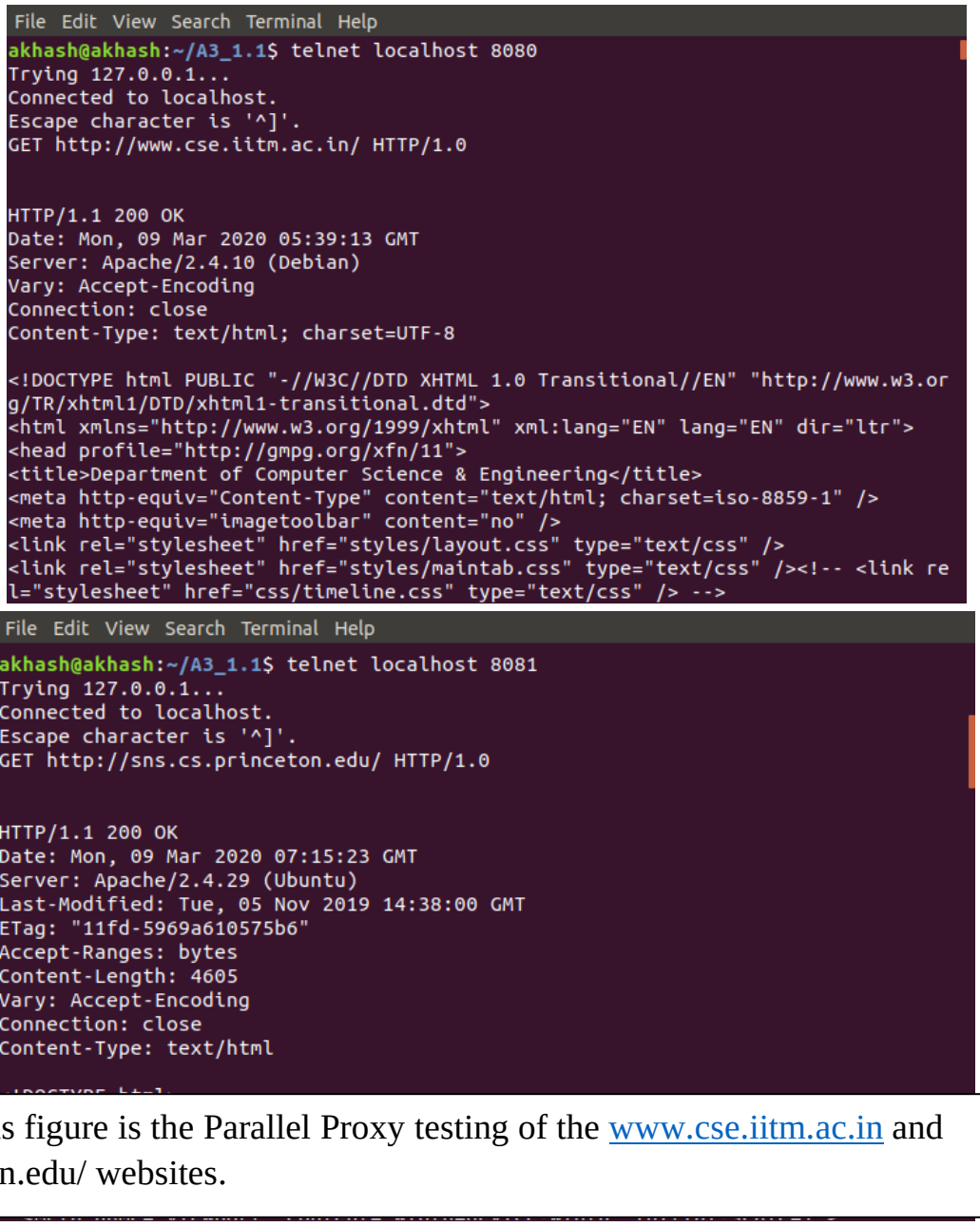
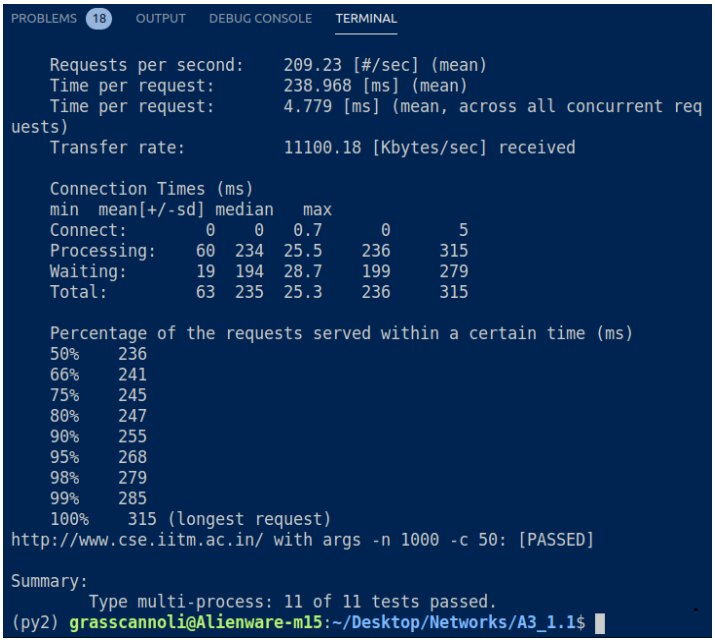


Figure 2.3: This figure is the Parallel Proxy testing of the www.cse.iitm.ac.in and sns.cs.princeton.edu/ websites.

c) Python Script Testers:



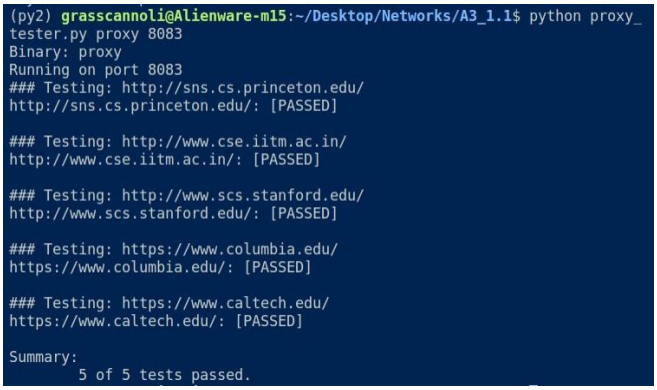
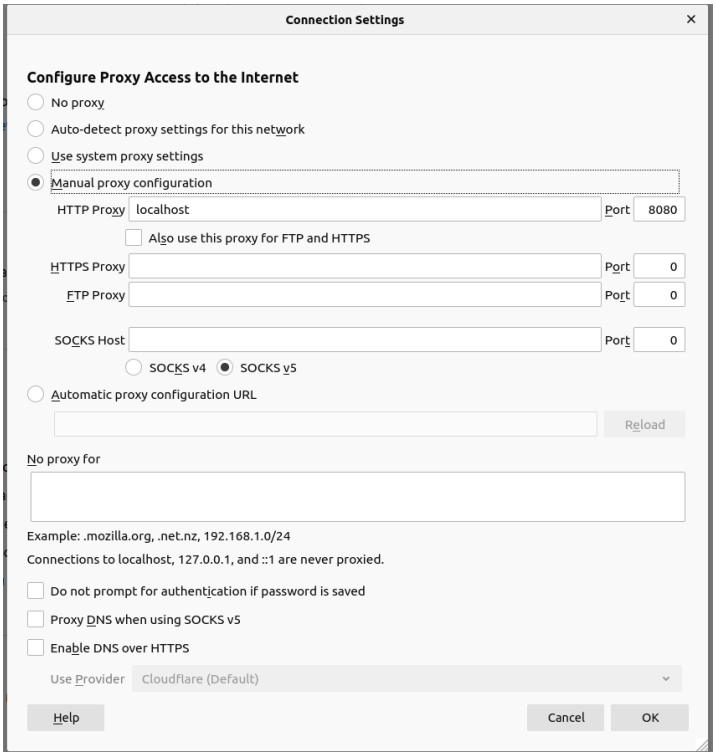


Figure 2.4: This figure is the testing output of both the proxy_tester_conc.py and proxy_tester.py scripts using the written "proxy.c" code.

d) Configuration of Mozilla Firefox:



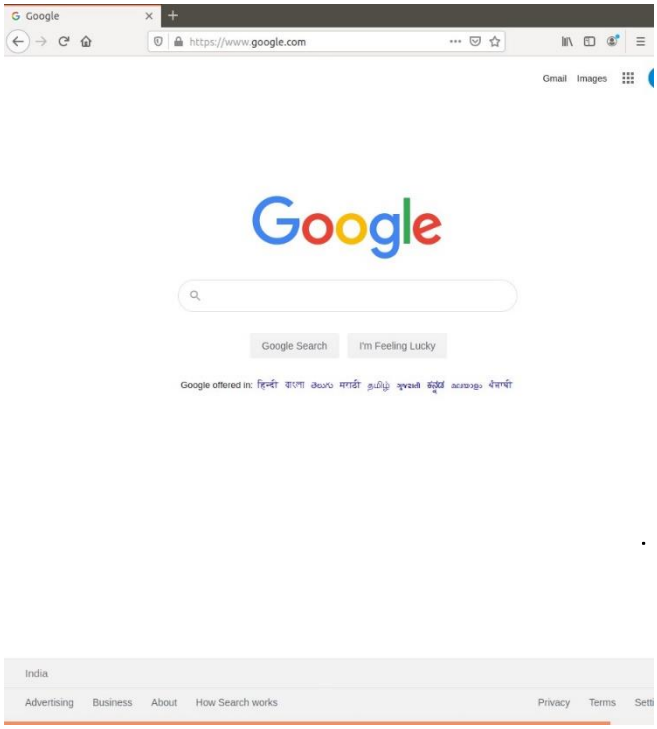


Figure 2.5: The first figure is the configuration settings of Mozilla Firefox using the written "proxy.c" code. The second figure shows the working of www.google.com.

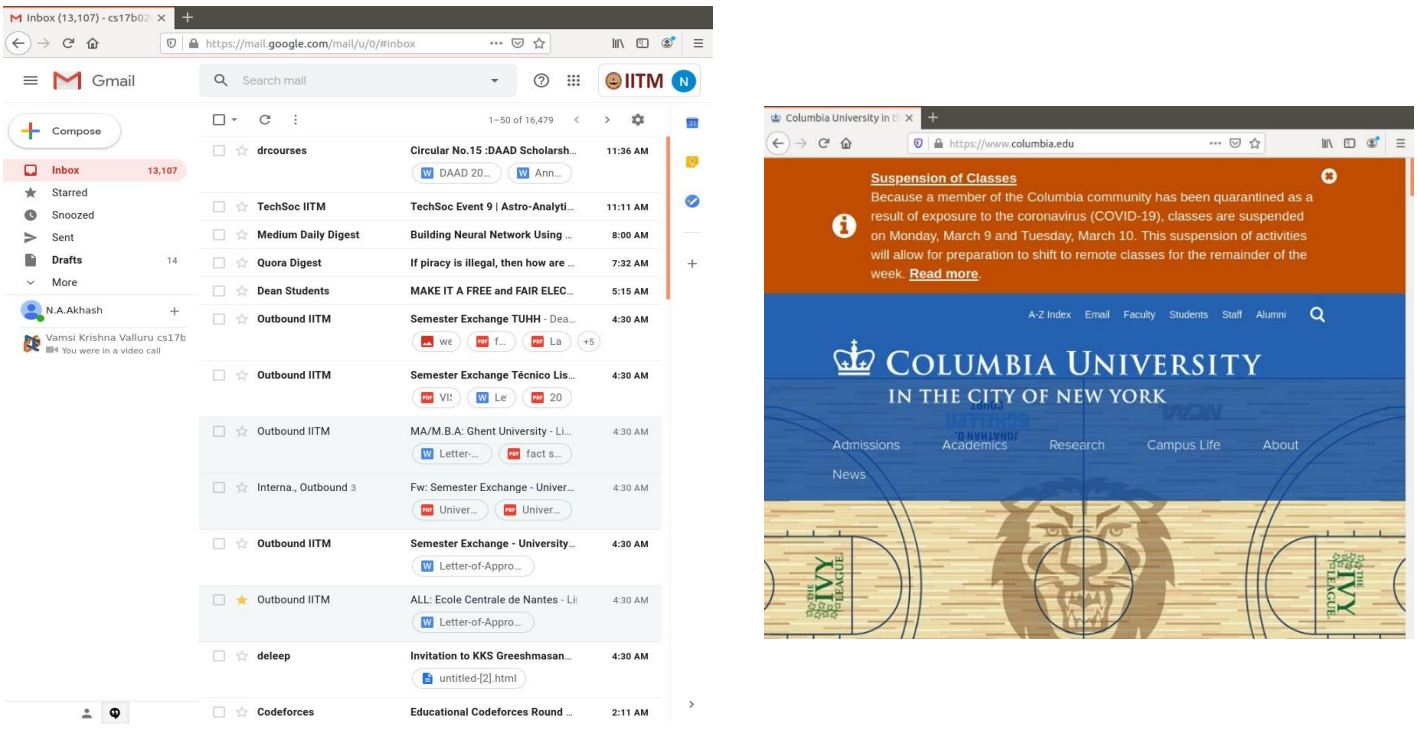


Figure 2.6: The first figure shows the working of www.gmail.com. The second figure shows the working of www.columbia.edu.

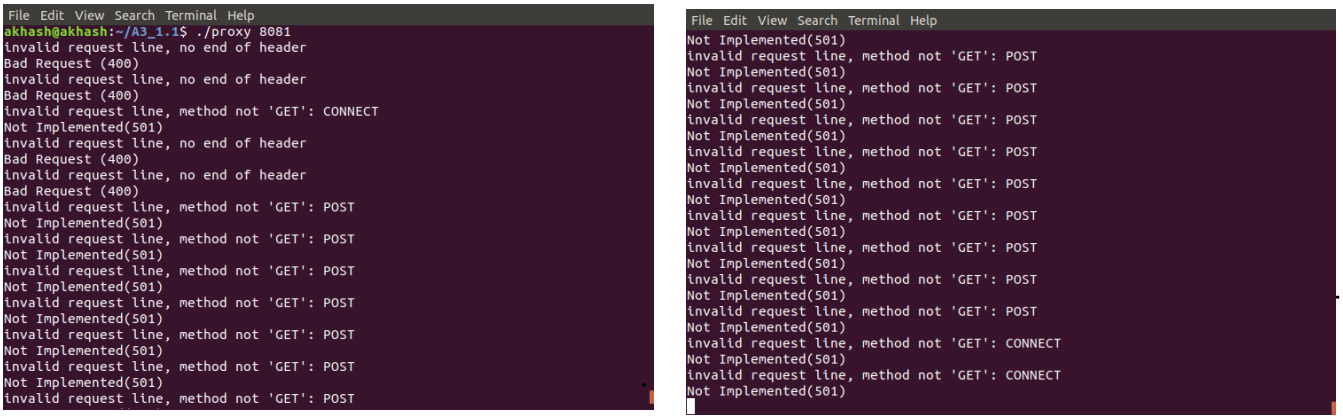


Figure 2.7: The first figure shows the working of www.gmail.com. The second figure shows the working of www.columbia.edu. Both of them show the terminal running with the proxy.

3) Learnings:

a) We Learnt how modularize and implement a network program.
b) Became comfortable with socket programming in C.
c) Understood how TCP and HTTP protocols work. Specifically, the GET method in HTTP.
d) Following the assignment, we have gained the confidence to tackle industry grade proxy server problems.

Acknowledgements:

a) Credits to Prof. Manikantan Srinivasan and his lectures/reference material.
b) Learnt to code TCP sockets in C and lookup URL IP from: https://www.cs.rpi.edu/moorthy/Courses/os98/Pgms/socket.html and https://gist.github.com/nolim1t/126991 respectively.
c) Acknowledge general discussion with Sriram V, Pranav R and Arnav M.